## REMARKS

Favorable reconsideration and allowance of the claims of the present application, as amended, is respectfully requested.

In the present Office Action, the Examiner first rejected Claims 29 and 31 under 35 U.S.C. §112, second paragraph, as having claim language allegedly unclear and indefinite.

Applicants have amended each of claims 29 and 31 to clarify that the "router queues " as claimed are associated with a first processing workstation (Main JVM as shown in Fig. 1 of the present application) which does not have any instantiators and, is shown as part of a cluster of workstations, however. The claims 29 and 31 have been clarified to distinguish the presences of instantiators at a second processing workstation, different from said first processing workstation yet part of the cluster.

Applicants submit that Claim 29 and 31 are now clear and definite and the Examiner is respectfully requested to withdraw the rejection under 35 U.S.C. §112, second paragraph.

In the present Office Action, the Examiner rejected Claims 29-31 and 33-34 under 35 U.S.C. §102(e) as allegedly anticipated by Broder et al. U.S. Patent No. 5,991,808 (hereinafter " Broder "), and further, rejected Claims 32 and 35 under 35 U.S.C. §103(a) as allegedly unpatentable over Broder.

The present invention as set forth in Claim 29 as now amended, is directed to a method of distributing work through a cluster of workstations for efficient distributed processing, said cluster having a plurality of workstations interconnected over a network, the method comprising:

receiving a work request at a first processing workstation;

classifying, at said first processing workstation, the work request into one or more tasks;

assigning said one or more tasks to one or more router queues, at said first processing workstation, capable of handling said one or more tasks;

dispatching said assigned one or more tasks for execution at a second processing workstation having an execution module residing therein, the execution module at said second processing workstation comprising one or more initiators for instantiating one or more objects to execute a work task, said initiators dynamically registering with a router to indicate readiness to accept work for processing, said objects instantiated by an initiator with a generic class name but having a different implementation specific to the workstation in which said initiator resides to enable use of system specific resources and enable a single version of an application to run on each workstation, said one or more router queues permitting work at different phases of completion to flow through said cluster of workstations; and

determining performance statistics associated with said one or more router queues, and, adding additional initiators to execute said one or more tasks based on the performance statistics of said one or more router queues.

Respectfully no new matter is being entered. For example, the subject matter of Claims 32 and 35 (now canceled) are now recited in amended Claim 29.

While the Examiner cites Broder at Col. 4, lines 21-28 (mainly) as allegedly teaching the receiving, classifying, and assigning of one or more tasks to one or more router queues, applicants respectfully disagree. Broder does not teach a cluster distinguished by the provision of a first processing workstation and a second processing workstation wherein the first

processing workstation is responsible for the receiving a work request, classifying, and assigning of one or more tasks of the request to one or more router queues, at said first processing workstation, and then dispatching of the tasks from the one or more queue routers to intantiators at the second workstations for task performance (object instantiation).

Moreover, Broder does not teach or suggest that the initiators dynamically register with a router to indicate readiness to accept work for processing as is taught in the specification at page 13 description of initiator registration with the router queues and now claimed in amended Claim 29.

Moreover, Broder does not teach or suggest the dispatching of the assigned one or more tasks from the first workstation for execution at the second processing workstation having an execution module residing therein, the execution module at said second processing workstation comprising one or more initiators for instantiating one or more objects to execute a work task, said initiators dynamically registering with a router to indicate readiness to accept work for processing, said objects instantiated by an initiator with a generic class name but having a different implementation specific to the workstation in which said initiator resides to enable use of system specific resources and enable a single version of an application to run on each workstation.

While the Examiner had cited Broder in the rejection of Claim 35 (now canceled) that in view of Broder it would be obvious to try, applicants respectfully disagree.

Broder, contrary to the Examiner's rejection, first of all does not distinguish between first and second workstations with dynamic assignment and queuing of tasks at the first workstation, and processing tasks at other second workstations by instantiators that dynamically register with the routing queues.

Moreover, the separation of receiving requests and classifying them as one or more tasks, with the tasks separately assignable for queuing on one or more routers and, provision of one or more second workstations having initiators that instantiate objects that processes the request at a second processing workstation is advantageous as the one or more router queues permit work at different phases of completion to flow through said cluster of workstations. This limitation added to Claim 29 is found in the present specification at page 11, lines 26 and is neither taught nor suggested by Broder.

Broder, while directed to a server cluster, teaches queuing within the individual server and optimizes processor tasks in a manner completely different than as claimed in amended Claim 29.

Moreover, the provision of second processing workstations comprising one or more initiators for instantiating one or more objects to execute a work task, the initiators dynamically registering with a router to indicate readiness to accept work for processing, said objects instantiated by an initiator with a generic class name but having a different implementation specific to the workstation in which said initiator resides to enable use of system specific resources and enable a single version of an application to run on each workstation is not obvious to try as each second processing station of the present invention may be different (comprise heterogeneous resources) and may not have the same object version and processing resources. To the contrary, Broder's server cluster comprise homogeneous resources.

It is important to the invention that each heterogeneous node comprises one or more initiators for instantiating one or more objects to execute a work task, the objects instantiated with a generic class name but having a different implementation specific to the node in which the initiator resides to enable use of system specific resources and enable a single

version of an application to run on each heterogeneous node.

Thus, in the present invention, there is no need for a configuration file that specifies special hardware/software features for the second workstation running a distributed work. In one embodiment, the present invention may take advantage of such special features if they exist. If the feature does not exist, the present invention uses the implementation code running on the second workstation. Moreover, the present invention as now claimed enables the same copy of an application to be run on any platform/version/operating system. The present invention includes the logic of special features in the classpath that is dynamically compiled with the work unit at run time. The classpath is unique to the workstation and requires no additional set up of this workload distribution invention.

Thus, key to the invention is that the work unit has the same class name in all the nodes, although their class path implementation may be different. Further, the workload manager software need not know of a special feature a particular workstation may include. The implementation advantages of a workstation are bound to the work unit at dynamic compilation time, allowing for transparency of a distributed system of workstations.

The present amendment to Claim 29 now sets forth these patentable features that are not taught by Broder. While the Examiner takes "Official Notice" that multiple objects having same class names may be implemented and is well known in the art - respectfully, Broder neither teaches nor suggests that these same named objects residing on different nodes have different implementations such that they are executable on different platforms/operating systems and yet can be invoked by a single common user application in the manner as in the present invention.

In view of the foregoing, Applicants respectfully submit that the amendments to

independent Claim 29 made herein obviate all the rejections based on 35 U.S.C. §102(e). In view of the amendments made to Claim 29, Claim 31 has been further amended. It is thus respectfully requested that the Examiner's rejections of these claims and all remaining claims dependent thereon, be withdrawn.

In view of the foregoing remarks herein, it is respectfully submitted that this application is in condition for allowance. Accordingly, it is respectfully requested that this application be allowed and a Notice of Allowance be issued. If the Examiner believes that a telephone conference with the Applicants' attorneys would be advantageous to the disposition of this case, the Examiner is requested to telephone the undersigned, Applicants' attorney, at the following telephone number: (516) 742-4343.

Respectfully submitted,

Steven Fischman
Registration No. 34,594

SCULLY, SCOTT, MURPHY & PRESSER, P.C.
400 Garden City Plaza, Suite 300
Garden City, New York 11530
(516) 742-4343

SF:gc